Sensors & Transducers
ISSN 1726- 5479

http://www.sensorsportal.com

Formal Presentation of Fuzzy Systems with Multiple Sensor Inputs

Alexander GEGOV, Dimitrios MAKETAS
University of Portsmouth, School of Computing
Buckingham Building, Portsmouth PO1 3HE, United Kingdom
Tel: +44-23-92846381, Fax: +44-23-92846364
E-mail: alexander.gegov@port.ac.uk, URL: www.port.ac.uk/csse

Abstract: The paper addresses the problems of complexity in fuzzy rule based systems with multiple sensor inputs. The number of fuzzy rules in this case is an exponential function of the number of inputs. Some of the existing methods for rule base reductions are reviewed and their drawbacks summarised. As an alternative, a novel methodology for complexity management in fuzzy systems is presented which is based on formal presentation techniques such as integer tables. A Matlab example is shown illustrating the presentation of a fuzzy rule base with an integer table. Finally, some future research directions are outlined within the framework of the proposed methodology.

Keywords: Multiple sensors, fuzzy systems, rule bases, complexity management, formal presentation.

---

1. Introduction

A fuzzy system is usually described by if-then rules of the form

$$\text{If } i_1 \text{ is } v_{i1,1} \text{ and } \ldots \text{ and } i_m \text{ is } v_{im,1} \text{ then } o_1 \text{ is } v_{o1,1} \text{ also } \ldots \text{ also } o_n \text{ is } v_{on,1} \tag{1}$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\text{If } i_1 \text{ is } v_{i1,r} \text{ and } \ldots \text{ and } i_m \text{ is } v_{im,r} \text{ then } o_1 \text{ is } v_{o1,r} \text{ also } \ldots \text{ also } o_n \text{ is } v_{on,r}$$

where m is the number of sensor inputs, n is the number of decision outputs and r is the number of fuzzy rules in the system. In this case, i_p, $p=1,m$ represents the p-th input, $v_{ip,s}$ $p=1,m$, $s=1,r$ is the linguistic value of the p-th input in the s-th rule, o_q, $q=1,n$ represents the q-th output and $v_{oq,s}$ $q=1,n$, $s=1,r$ is the linguistic value of the q-th output in the s-th rule.

Generally, the number and the meaning of the linguistic values that each input can take vary as inputs usually have different crisp variation ranges and specific crisp physical meanings. However, for simplicity this peculiarity is not reflected explicitly in the rule base represented by Equation (1).

The number of rules in a fuzzy system r is an exponential function of the number of the inputs m and the number of linguistic values k that these inputs can take. In most cases, this exponential function is in the form

$$r = k^m \tag{2}$$

It is obvious from Equation (2) that for a fuzzy system with 2 inputs which can take 5 linguistic values the number of rules will be 25. However, for the case of 3 inputs the number of rules becomes 125 and it is not difficult to imagine what the impact on the number of rules would be of 4, 5 or even more inputs.

The above presented considerations show the high level of computational complexity of fuzzy systems with multiple sensor inputs even for the case of a fairly small number of inputs. Bearing in mind that many real life systems are usually characterised by a much larger number of inputs and often have to be operated in real-time, it is obvious that the resulting computational complexity has to be taken seriously. Moreover, it has been shown that efficient algorithms contribute to a much greater extent to the timely solution of large scale problems than fast computers do.

It must be noted that the number of rules in a fuzzy system is only a rough estimate of its computational complexity. The actual complexity is a function of the number of rules as most of the major steps in a fuzzy system such as fuzzification, inference and defuzzification are dependent on this number. However, for the purpose of complexity management it is usually sufficient to reduce the number of fuzzy rules and this is what most of the known methods for complexity reduction are focused on.

2. Rule Base Reduction Methods

Most of the available methods for complexity reduction in fuzzy systems reduce the number of fuzzy rules by either reducing the number of inputs or the number of linguistic values that these inputs can take [1]. These methods are classified in three groups and are briefly discussed below.

The first group of methods for complexity reduction are aimed at either removing less significant linguistic values or merging similar linguistic values [2]. For example, if the original fuzzy system is described by the five linguistic values *very small* (VS), *small* (S), *medium* (M), *big* (B) and *very big* (VB) then the values VS and VB could be removed in which case the values S and B will cover all cases that would otherwise be classified as VS and VB, respectively. Alternatively, the linguistic values VS and S could be merged into a new value called *fairly small* (FS) whereas B and VB could be merged into the corresponding new value *fairly big* (FB). Although these methods are very easy to apply, the process of removing or merging linguistic values is usually associated with loss or aggregation of information and this can have a negative impact on the ability of the reduced fuzzy system to adequately represent the original system.

The second group of methods for complexity reduction are aimed at either removing less significant inputs or fusing similar inputs [3]. For example, if the original fuzzy system is described by the inputs *position* (P), *velocity* (V) and *acceleration* (A) then the input A could be removed. Alternatively, the inputs V and A could be fused into a new input called *velocity / acceleration* (VA). Although these methods are also quite easy to apply, the process of removing or fusing inputs may actually lead to undesirable and risky simplification whereby the reduced fuzzy system is too different from the original system.

The third group of methods for complexity reduction are aimed at either rearranging or transforming the inputs, the linguistic values or the rules in the original fuzzy system in a way that leads to the reduction of the number of rules. Some of the methods that belong to this group are discussed separately below.

The most popular from the third group is the so-called *hierarchical* method which represents the original fuzzy system as a multilayer hierarchical structure of subsystems such that each layer (subsystem) has only two inputs and one output [4]. For example, a system with three inputs i_1, i_2 and i_3 can be represented as two cascaded subsystems where i_1 and i_2 are the inputs to the first subsystem. In this case, the corresponding output o_1 from the first subsystem is used as an input to the second subsystem together with the remaining input to the original system i_3. In other words, if the inputs can take 3 linguistic values then the original fuzzy system will have $3^3 = 27$ rules whereas the reduced system will have only $3^2 + 3^2 = 18$ rules. Although this method has become quite popular recently, it has some significant drawbacks such as the arbitrary selection of the subsystems and the potentially big number of layers which to some extent outweigh the positive effect from the reduced number of rules.

Two other methods from the third group that have gained relatively high popularity recently are the so-called *SVD* and *COMB* methods. They are both based on the idea of approximating the behaviour of the original fuzzy system with a reduced fuzzy system.

The *SVD* method uses a singular value decomposition of the matrix representing the actual values of the outputs from the original fuzzy system as a result of which the corresponding number of linguistic values for the inputs is reduced [5]. For example, a system with 2 inputs that can take 5 linguistic values such as *negative big* (NB), *negative small* (NS), *zero* (Z), *positive small* (PS) and *positive big* (PB) can be approximated with a system with the same number of inputs and only two linguistic values such as *negative* (N) and *positive* (P). However, the application of this method to a system with more than two inputs is computationally complex and not worth the effort.

The *COMB* method is based on the fact that some conjunctive rule bases can be represented into an almost equivalent disjunctive form [6]. For example, a system with 2 inputs and 1 output in the conjunctive format *if (i_1 and i_2) then o_1* can be represented in the disjunctive format *if (i_1 then o_1) or if (i_2 then o_2)*. In this case, a system with 2 inputs that can take 5 linguistic values is represented by 25 rules but as a result of the application of this method can be reduced to a system with the same number of inputs and linguistic values that these inputs can take but the corresponding number of rules will be only 10. However, this method can only be used for a special class of systems called *additively separable* and therefore it has a very limited application scope.

It follows from the detailed considerations above that most of the available methods for complexity reduction in fuzzy systems have serious drawbacks such as limited application scope and empirical nature. Therefore, it is crucial to find a more universal and systematic complexity management methodology that will provide a better solution to the problem.

3. Complexity Management Methodology

The novel methodology introduced here is based on the idea of formal presentation of the fuzzy rule based systems. It builds on earlier works by some of the authors on relational decoupling [7] and more recent works on linguistic decoupling [8]. The main advantages of this methodology in comparison to previous works of the authors and similar works by others are that:

- it is widely applicable irrespective of the properties of the fuzzy system,
- it lends itself easily to formalisation and mathematical manipulation.

The underlying philosophy of this novel approach deals with complexity related problems in fuzzy systems not only by reducing the number of fuzzy rules but from a much wider perspective that taking into account other factors that contribute to this complexity such as the number of inputs and outputs in the rules. For this reason, the more general term *complexity management* is used here in stead of the relatively specific term *complexity reduction*.

Other advantages of this methodology in comparison to existing methods are the following:

- it does not require any underlying knowledge about the associated physical process,
- it provides an equivalent presentation of the behaviour of the initial fuzzy system,
- it can be applied to a fuzzy system with an arbitrary number of inputs and outputs,
- it can be applied to a fuzzy system with an arbitrary type of rule base.

The complexity management methodology presented here takes into account the properties of the fuzzy system rule base. These properties reflect the existence of all permutations of linguistic values of inputs and outputs as well as the type of mapping between the permutations of linguistic values of the inputs in the *if* part of the rule base and the corresponding permutations of linguistic values of the outputs in the *then* part. In particular, there are four basic properties of a fuzzy rule base and they are given in the definitions below.

Definition 1
A fuzzy rule base is *complete* if and only if all possible permutations of linguistic values of the inputs are present in the *if* part of the rule base.

Definition 2
A fuzzy rule base is *exhaustive* if and only if all possible permutations of linguistic values of the outputs are present in the *then* part of the rule base.

Definition 3
A fuzzy rule base is *consistent* if and only if every available permutation of linguistic values of the inputs is mapped onto only one available permutation of linguistic values of the outputs.

Definition 4
A fuzzy rule base is *monotonic* if and only if every available permutation of linguistic values of the outputs is mapped from only one available permutation of linguistic values of the inputs.

4. Formal Presentation of Rule Bases with Integer Tables

Integer tables allow fuzzy systems to be formally presented in a more compact form [9]. However, the process of converting a rule base into an integer table is usually ignored and this makes the fuzzy system under consideration less transparent and more difficult for interpretation. To overcome this drawback, a detailed algorithm has been designed which is shown in Figure 1.

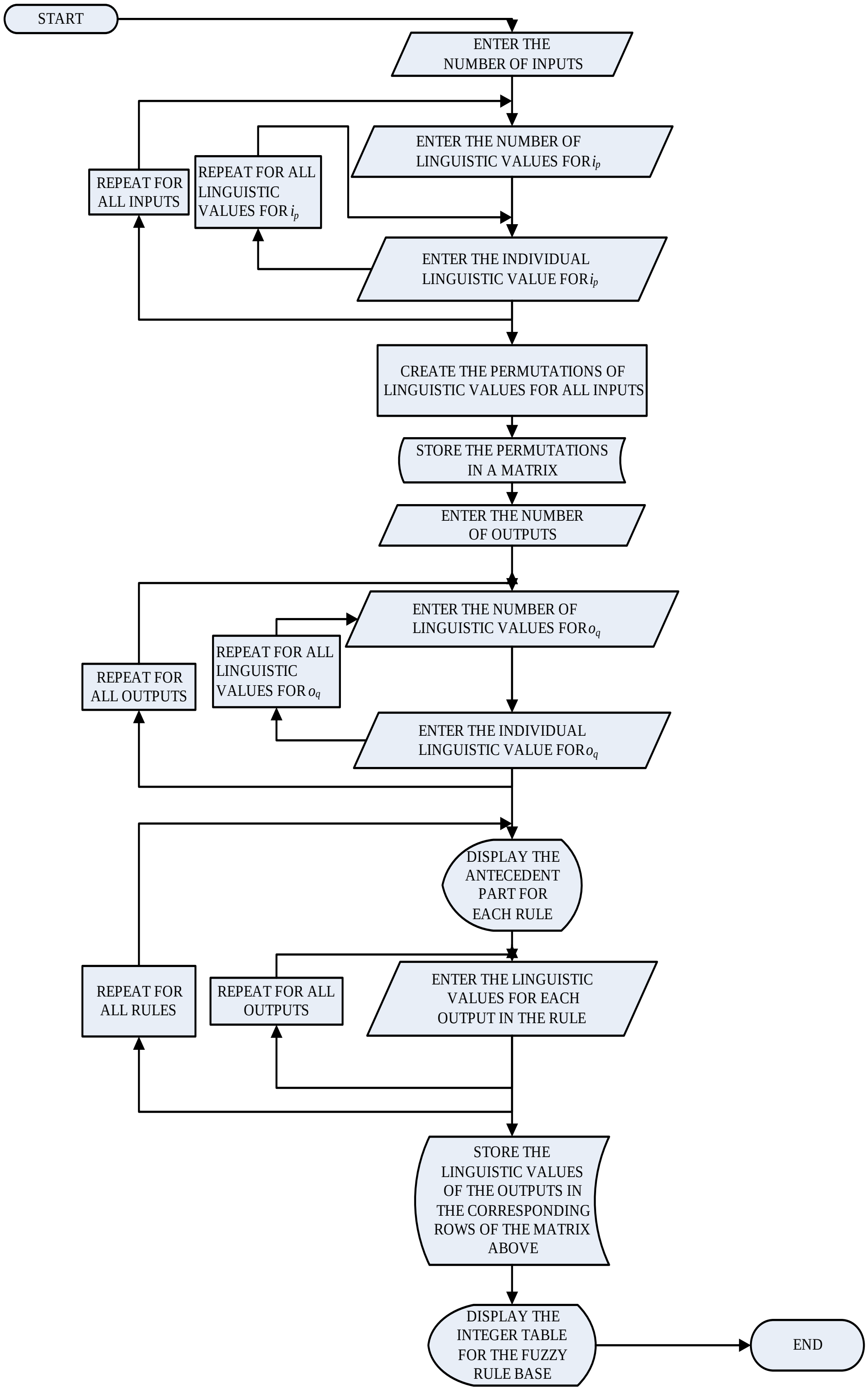

Fig. 1. *Algorithm for Converting a Rule Base into an Integer Table*

For consistency, all terms and notations for the algorithm in Figure 1 are the same as the ones used for Equation (1). The algorithm is based on a dialog with the user who is prompted to enter all the information about the rule base that is required for its conversion into an integer table.

The algorithm above has also been implemented in Matlab and an example is shown below for illustration purposes. The example is about an aircraft landing control system with inputs i_1 (Height) and i_1 (Velocity), and output o_1 (Force) [10]. The fuzzy rule base for this system is given in a look-up table format which shown in Table 1.

Table 1. *Look-up table for the rule base of the aircraft landing control system*

	i_2	**DL**	**DS**	**Z**	**US**	**UL**
i_1		o_1	o_1	o_1	o_1	o_1
NZ	o_1	UL	UL	Z	DS	DS
S	o_1	UL	US	Z	DS	DL
M	o_1	US	Z	DS	DL	DL
L	o_1	Z	DS	DL	DL	DL

The linguistic values for the input i_1 in the look-up table are NZ (Near Zero), S (Small), M (Medium), L (Large) and they are shown in bold as row labels. The linguistic values for the input i_2 are DL (Down Large), DS (Down Small), Z (Zero), US (Up Small), UL (Up Large) and they are also shown in bold but as column labels. The linguistic values for the output o_1 are the same as the ones for the input i_2 and they are shown in normal font as elements of the look-up table.

The rule base from the look-up table above is presented by if-then rules which are shown in Equation (3).

(3)

Rule 1: If i_1 is NZ and i_2 is DL then o_1 is UL

Rule 2: If i_1 is NZ And i_2 is DS Then o_1 is UL

Rule 3: If i_1 is NZ Aand i_2 is Z then o_1 is Z

Rule 4: If i_1 is NZ and i_2 is US then o_1 is DS

Rule 5: If i_1 is NZ and i_2 is UL then o_1 is DS

Rule 6: If i_1 is S and i_2 is DL then o_1 is UL

Rule 7: If i_1 is S and i_2 is DS then o_1 is US

Rule 8: If i_1 is S and i_2 is Z then o_1 is Z

Rule 9: If i_1 is S and i_2 is US then o_1 is DS

Rule 10: If i_1 is S and i_2 is UL then o_1 is DL

Rule 11: If i_1 is M and i_2 is DL then o_1 is US

Rule 12: If i_1 is M and i_2 is DS then o_1 is Z

Rule 13: If i_1 is M and i_2 is Z then o_1 is DS

Rule 14: If i_1 is M and i_2 is US then o_1 is DL

Rule 15: If i_1 is M and i_2 is UL then o_1 is DL

Rule 16: If i_1 is L and i_2 is DL then o_1 is Z

Rule 17: If i_1 is L and i_2 is DS then o_1 is DS

Rule 18: If i_1 is L and i_2 is Z then o_1 is DL

Rule 19: If i_1 is L and i_2 is US then o_1 is DL

Rule 20: If i_1 is L and i_2 is UL then o_1 is DL

In order to convert the rule base above into an integer table, it is necessary to code all linguistic values as positive integers whereby the linguistic values for each input and output are first sorted in an increasing order. As a result, the linguistic values of the input i_1 are coded as NZ=1, S=2, M=3, L=4 whereas the linguistic values for the input i_2 and the output o_1 are mapped as DL=1, DS=2, Z=3, US=4, UL=5. This coding allows the rule base from Equation (3) to be converted into an integer table which is shown in Table 2.

Table 2. *Integer table for the rule base of the aircraft landing control system*

Rule number	**Input i_1**	**Input i_2**	**Output o_1**
1	1	1	3
2	1	2	2
3	1	3	1
4	1	4	1
5	1	5	1
6	2	1	4
7	2	2	3
8	2	3	2
9	2	4	1
10	2	5	1
11	3	1	4
12	3	2	3
13	3	3	2
14	3	4	1
15	3	5	1
16	4	1	3
17	4	2	2
18	4	3	1
19	4	4	1
20	4	5	1

The integer table in Table 2 shows clearly that the information from the rule base in Equation (3) has been significantly compressed whereby all unnecessary details have been removed. This type of compression reduces the complexity of the fuzzy system without compromising its quality.

5. Conclusions

The complexity management methodology introduced in this paper provides a good basis for further research in fuzzy systems with multiple sensor inputs. The integer tables can be used for the development of more advanced formal presentation techniques such as Boolean matrices and binary relations. These techniques can be further used for formal manipulation of complex fuzzy systems for the purpose of their analysis and synthesis. The proposed methodology also opens new horizons for the study of multiple rule based fuzzy systems, e.g. fuzzy systems whose rule bases are interconnected within a grid type of structure.

Acknowledgements

The Matlab illustration example for formal presentation of fuzzy systems with integer tables has been done as part of a Final Year Project on the MSc Software Engineering Course at the University of Portsmouth.

References

[1]. M.Jamshidi, *Large Scale Systems: Modelling, Control and Fuzzy Logic.* Prentice-Hall, 1997.
[2]. M.Setnes, *Fuzzy Rule Base Simplification Using Similarity Measures.* PhD Thesis, Delft University of Technology, 1995.
[3]. V.Lacrose, *Complexity Reduction of Fuzzy Controllers: Application to Multivariable Control.* PhD thesis, Toulouse Laboratory for Systems Analysis and Architecture, 1997.
[4]. G.Raju, J.Zhou, R.Kisner. Hierarchical fuzzy control, *International Journal of Control*, 54 (5) (1991), p.1201-1216.
[5]. Y.Yam. Fuzzy approximation via grid point sampling and singular value decomposition, *IEEE Transactions on Systems, Man and Cybernetics – Part B*, 27 (6) (1997), p.933-951.
[6]. W.Combs, J.Andrews. Combinatorial rule explosion eliminated by a fuzzy rule configuration, *IEEE Transactions on Fuzzy Systems*, 6 (1) (1998), p.1-11.
[7]. A.Gegov, *Distributed Fuzzy Control of Multivariable Systems.* Kluwer, 1996.
[8]. A.Gegov, R.Babuska, H.Verbruggen, Linguistic Analysis of Interactions in MIMO fuzzy systems. In *Proceedings of IFAC World Congress*, Beijing, China, 1999, p.249-254.
[9]. *Fuzzy Logic Toolbox User's Guide*, The MathWorks, 2001.
[10]. T.Ross, *Fuzzy Logic with Engineering Applications.* John Wiley & Sons, 2004.